Abstract:

Project is a program that differentiates between various objects such as a person, bags, car, glass, dog etc. Furthermore it counts the number of entities for each object. To identify the objects is really a complex problem.

Introduction:

To identify objects we need a model and I'm also using one named as Yolo object detector. First Yolo returns confidence and class ids for each object, then on the basis of confidence one can make decisions and lastly I created a hashmap for objects using a python dictionary and display results on image.

Background:

As I described in the introduction section I implemented a Yolo detector which trained on Coco datasets and coco dataset has almost eighty classes. The COCO dataset consists of 80 labels, including, but not limited to:

- People
- Bicycles
- Cars and trucks
- Airplanes
- Stop signs and fire hydrants
- Animals, including cats, dogs, birds, horses, cows, and sheep, to name a few
- Kitchen and dining objects, such as wine glasses, cups, forks, knives, spoons, etc.
- *…and much more!*

You can find a full list of what YOLO trained on the COCO dataset can detect using this link.

Approach:

When it comes to deep learning-based object detection, there are three primary object detectors you'll encounter:

- R-CNN and their variants, including the original R-CNN, Fast R- CNN, and Faster R-CNN
- Single Shot Detector (SSDs)
- YOLO

Why did I use YOLO? Because It trained on a large dataset and it is faster than CNN. So I can detect many objects in extremely low time.I implemented my solution in opencv for python. First I have to provide images to my program then it detects objects and returns class ids and confidence for each object. On the basis of confidence I put labels for each class in a hashmap and output result on image.

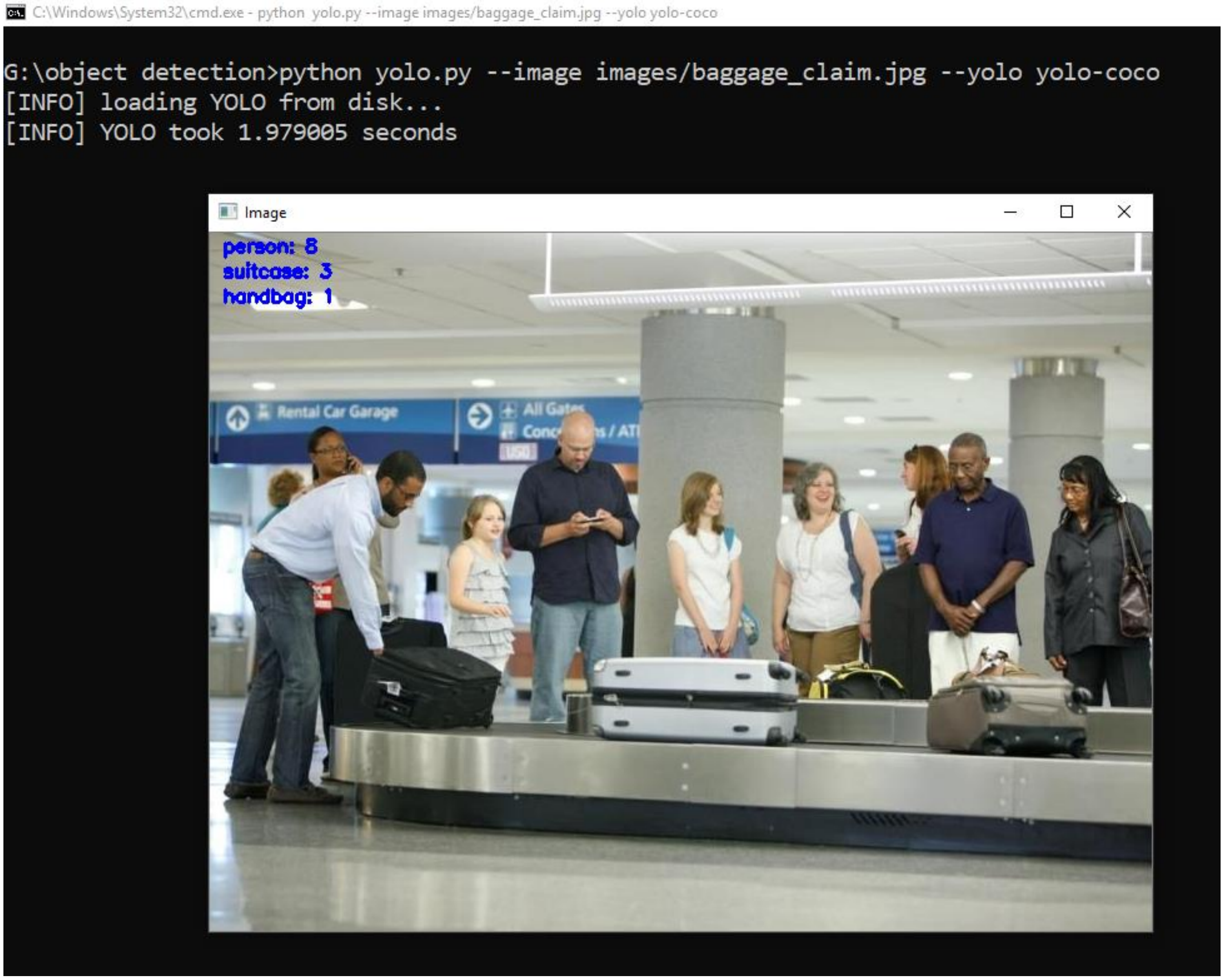
Results:

Following is the project structure

- Images directory (contains test images)
- Yolo-coco directory (contains detector configurations)
- yolo.py (program)

Command to execute program: `python yolo.py --image images/baggage_claim.jpg --yolo yolo-coco`

Following are the screenshots of results

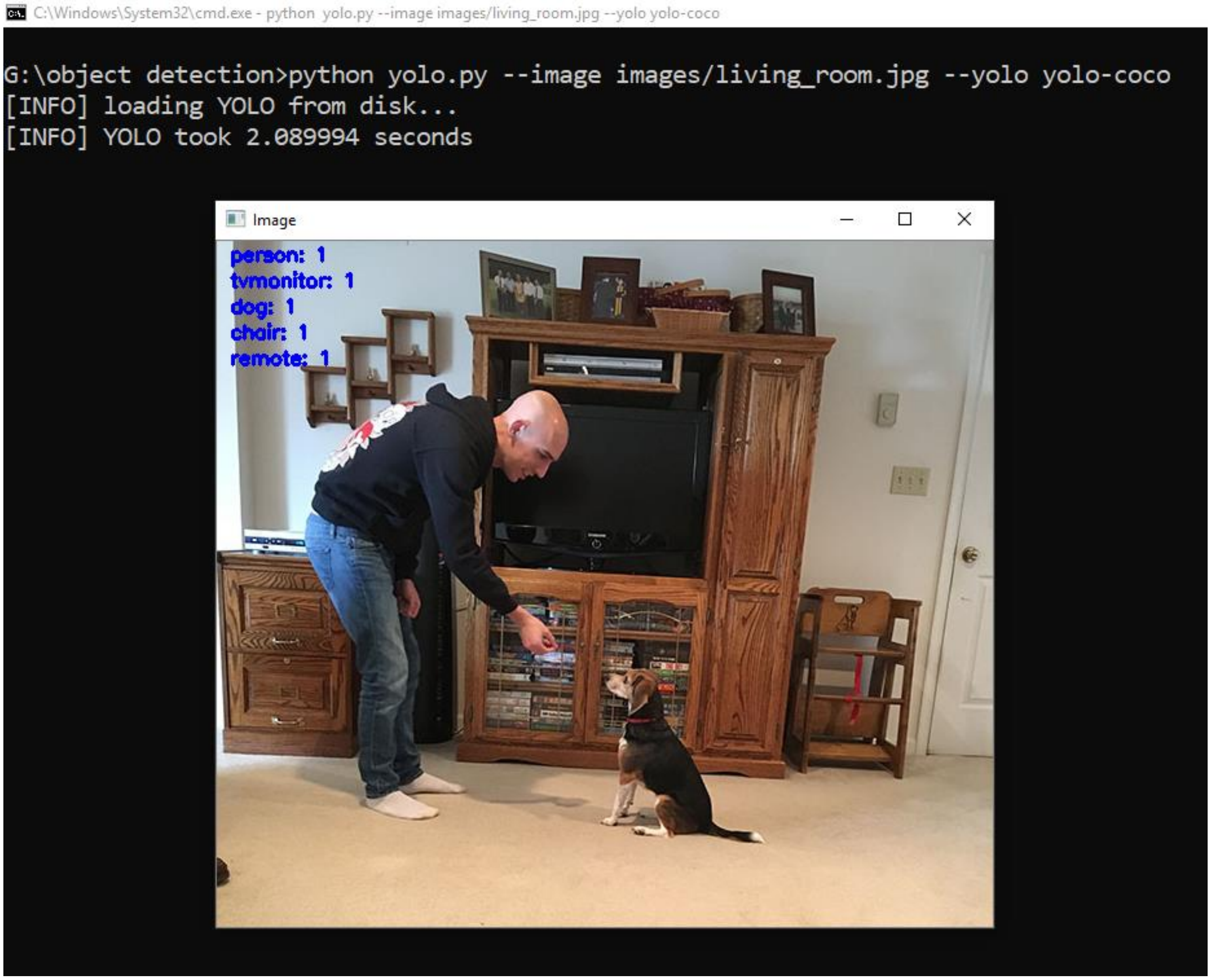

C:\Windows\System32\cmd.exe - python yolo.py --image images/baggage_claim.jpg --yolo yolo-coco
Image
Rental Car Garage
All Gates
Concessions / ATM

```
G:\object detection>python yolo.py --image images/baggage_claim.jpg --yolo yolo-coco
[INFO] loading YOLO from disk...
[INFO] YOLO took 1.979005 seconds
```

```
person: 8
suitcase: 3
handbag: 1
```

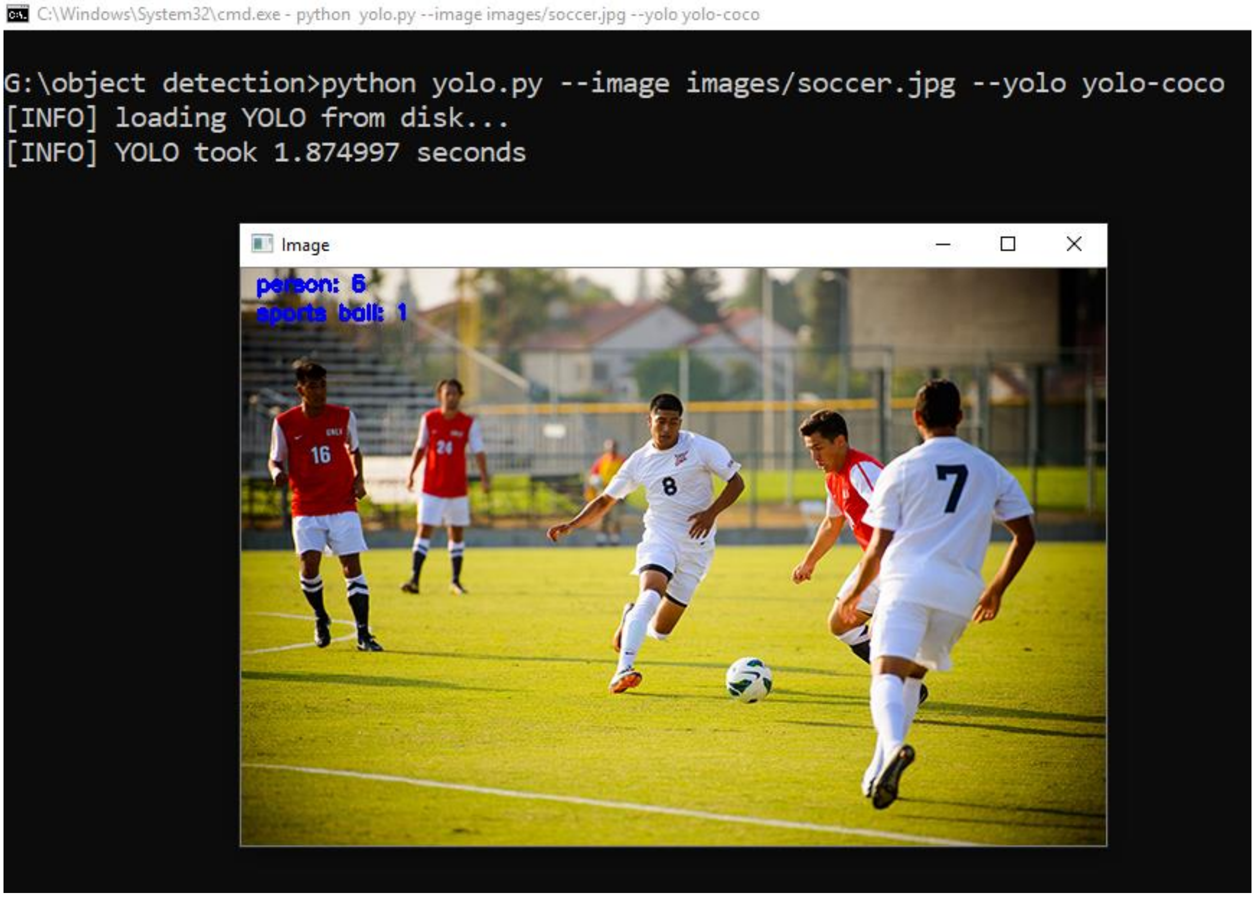
In the above image you can see that there are eight persons, three suitcases and one handbag. Yolo took 1.9 seconds to identify all these objects.

In the below image you can see a person holding a remote on his hand with a dog and a tv monitor. I'm particularly impressed that YOLO was able to detect the chair given that it's handmade, old fashioned "baby high chair".


C:\Windows\System32\cmd.exe - python yolo.py --image images/living_room.jpg --yolo yolo-coco
Image
person: 1
tvmonitor: 1
dog: 1
chair: 1
remote: 1

```
G:\object detection>python yolo.py --image images/living_room.jpg --yolo yolo-coco
[INFO] loading YOLO from disk...
[INFO] YOLO took 2.089994 seconds
```

In the below image YOLO is able to correctly detect each of the players on the pitch, including the soccer ball itself. Notice the person in the background who is detected despite the area being highly blurred and partially obscured.



C:\Windows\System32\cmd.exe - python yolo.py --image images/soccer.jpg --yolo yolo-coco
Image
person: 6
sports ball: 1

```
G:\object detection>python yolo.py --image images/soccer.jpg --yolo yolo-coco
[INFO] loading YOLO from disk...
[INFO] YOLO took 1.874997 seconds
```

LIST OF WORK:

Group members and contribution:

Saad Khan 101017608
Work: Performed 40% of the work on the project

Ashad Khan 101046422
Work: Performed 30% of the work on the project

Jason Yang 101028952
Raj Waidyaratne 100967063
The rest of the work was performed equally by the team members

Github Repository: https://github.com/ashadak/COMP4102_FinalProject-.git